Website fingerprinting attacks against Tor Browser Bundle: a comparison between HTTP/1.1 and HTTP/2

T.T.N. Marks BSc.
K.C.N. Halvemaan BSc.

University of Amsterdam

System and Network Engineering
Research Project #1

February 8, 2017

Overview

Introduction

1. **Tor:** The second generation onion router
2. "Tor is free software and an open network that helps you **defend against traffic analysis**, a form of network surveillance that threatens personal freedom and privacy, confidential business activities and relationships, and state security."[1]
3. Often used as part of the Tor Browser Bundle (TBB).

[1] https://www.torproject.org/, retrieved on 2017-02-02.

Problem statement

1. Website fingerprinting possible despite encryption and obfuscation techniques.
2. An eavesdropper might learn which website you have visited based on the meta data of the encrypted TCP/IP stream.
3. The web is moving from HTTP/1.1 to HTTP/2, what does this mean for website fingerprinting?
4. HTTP/2 still disabled in the TBB by default because code is not audited and possible security implications are unclear.

Research questions

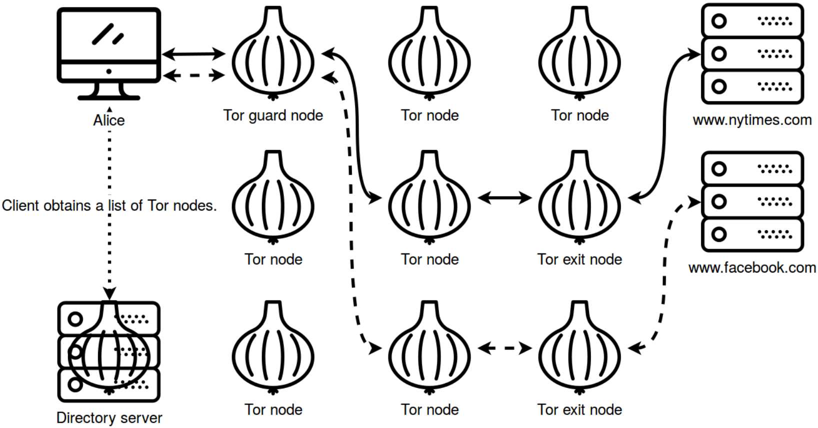
1. Can a website fingerprinting attack be done on a TBB enabled with HTTP/2?
2. Is there a difference in website fingerprinting attacks on a TBB enabled with just HTTP/1.1 and a TBB enabled with HTTP/2?

What is new in HTTP/2?

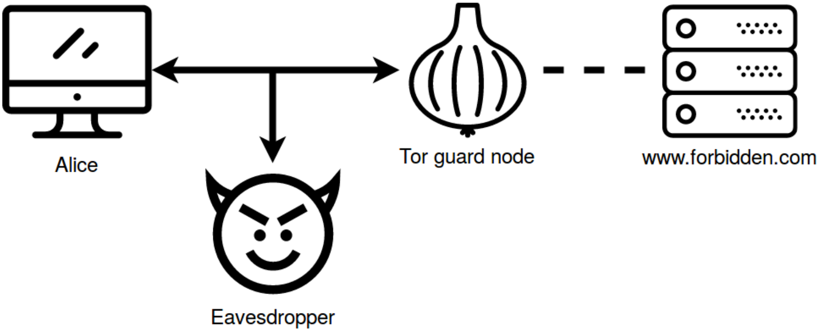
1. Mandatory HTTPS in all major browsers (de facto standard[2]).
2. Data compression of HTTP headers.
3. **Prioritisation of requests**.
4. **Multiplexing multiple requests over a single TCP/IP connection**.

[2] `https://http2.github.io/faq/#does-http2-require-encryption`, retrieved on 2017-02-03.

How Tor works.



Website fingerprinting



Related work

1. Fingerprinting encrypted HTTP traffic (Liberatore and Levine, 2006).

Related work

1. Fingerprinting encrypted HTTP traffic (Liberatore and Levine, 2006).
2. Extended to Tor by Herrmann et al. (2009).

Related work

1. Fingerprinting encrypted HTTP traffic (Liberatore and Levine, 2006).
2. Extended to Tor by Herrmann et al. (2009).
3. Improved by Panchenko et al. (2011) by using a Support Vector Machine.

Related work

1. Fingerprinting encrypted HTTP traffic (Liberatore and Levine, 2006).
2. Extended to Tor by Herrmann et al. (2009).
3. Improved by Panchenko et al. (2011) by using a Support Vector Machine.
4. Various defenses were discussed by Cai et al. (2012), of which the 'padding defense' was implemented in Tor.

Related work

1. Fingerprinting encrypted HTTP traffic (Liberatore and Levine, 2006).
2. Extended to Tor by Herrmann et al. (2009).
3. Improved by Panchenko et al. (2011) by using a Support Vector Machine.
4. Various defenses were discussed by Cai et al. (2012), of which the 'padding defense' was implemented in Tor.
5. A review of earlier methods was given in Wang and Goldberg (2013), their results were better but unrealistic setting.

Related work

1. Fingerprinting encrypted HTTP traffic (Liberatore and Levine, 2006).
2. Extended to Tor by Herrmann et al. (2009).
3. Improved by Panchenko et al. (2011) by using a Support Vector Machine.
4. Various defenses were discussed by Cai et al. (2012), of which the 'padding defense' was implemented in Tor.
5. A review of earlier methods was given in Wang and Goldberg (2013), their results were better but unrealistic setting.
6. The previous work on Tor was done by looking at HTTP/1.1 traffic.

Overview

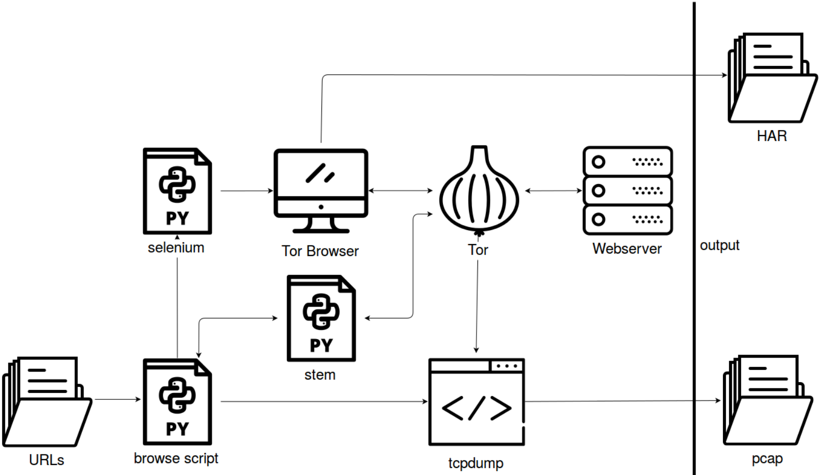
Overall Implementation

1. Get a list of websites supporting HTTP/2.
2. Visit each website 40 times in TBB for both HTTP/1.1 and HTTP/2:
    1. Make packet capture and save corresponding HTTP Headers.
    2. Convert packet captures to "traces".
3. Calculate distance between traces.
4. Use distances to train a SVM and use it to predict unseen traces.

URLs

1. Alexa top million websites of 2017-01-14.
2. Test top 5000 with curl for HTTP/2 responses.
3. 1110 of 5000 websites were HTTP/2 capable.
4. All Google TLDs were removed, except "google.com".
5. Top 130 of the HTTP/2 enabled websites were retrieved.

Setup


HAR
selenium
Tor Browser
Tor
Webserver
output
stem
URLs
browse script
tcpdump
pcap
PY
PY
PY

Problems after scraping

1. Invalid captures, that were removed from our sample.
    1. Websites redirecting to plain `http://`.
    2. Websites using Cloudflare, as they would show a captcha screen by default.
    3. Websites that failed to load completely more than 25% of the time.
2. Left us with 56 of 130 websites scraped.

Converting packet captures to traces

1. Based on method by Wang and Goldberg (2013).
2. Check HTTP Archive (HAR) content and verify HTTP version and status OK.
3. Filter out retransmitted and out-of-order TCP/IP packets.
4. One or more Tor cells in TCP/IP packet, extracted by rounding length of data in bytes to nearest multiple of 512 and dividing by 512.
5. Direction indicated with sign: negative for incoming and positive for outgoing.
6. Resulting trace is a list of only 1's and -1's indicating the direction, order and frequency of Tor cells for a specific website.
7. Still some "noise" left in traces due to SENDME Tor cells.

Training the SVM

1. Distance between traces calculated with the *optimal string aligment distance* (Wang and Goldberg, 2013).
   1. Took about four hours to compute on the DAS5 supercomputer using 10 nodes (Bal et al., 2016).
2. Train and test the SVM in closed world model.
   1. 36 training cases and 4 testing cases for each site.
   2. 10-fold cross validation with one accuracy value for each of the folds, so 10 accuracy's per tested set.

Results

Train \ Test	HTTP/1.1	HTTP/2
HTTP/1.1	$\boldsymbol{\overline{x} = 88.036\%}$ $\boldsymbol{s = 2.0164\%}$	$\overline{x} = 64.687\%$ $s = 6.6631\%$
HTTP/2	$\overline{x} = 54.667\%$ $s = 3.5286\%$	$\boldsymbol{\overline{x} = 86.485\%}$ $\boldsymbol{s = 3.0871\%}$

Results

Train \ Test	HTTP/1.1	HTTP/2
HTTP/1.1	$\boldsymbol{\overline{x} = 88.036\%}$ $\boldsymbol{s = 2.0164\%}$	$\overline{x} = 64.687\%$ $s = 6.6631\%$
HTTP/2	$\overline{x} = 54.667\%$ $s = 3.5286\%$	$\boldsymbol{\overline{x} = 86.485\%}$ $\boldsymbol{s = 3.0871\%}$

1. HTTP/1.1 by Wang and Goldberg (2013): $\overline{x} = 90\%$ $s = 6\%$

Results

Train \ Test	HTTP/1.1	HTTP/2
HTTP/1.1	$\boldsymbol{\overline{x} = 88.036\%}$ $\boldsymbol{s = 2.0164\%}$	$\overline{x} = 64.687\%$ $s = 6.6631\%$
HTTP/2	$\overline{x} = 54.667\%$ $s = 3.5286\%$	$\boldsymbol{\overline{x} = 86.485\%}$ $\boldsymbol{s = 3.0871\%}$

1. HTTP/1.1 by Wang and Goldberg (2013): $\overline{x} = 90\%$ $s = 6\%$
2. Paired t-test of accuracy's between the HTTP/1.1 and HTTP/2 sets: $p_{\text{value}} = 0.19392$, with $\alpha = 0.05$.
   The difference is *not* statistically significant: $p_{\text{value}} > \alpha$.

Conclusion

1. It is possible to do a website fingerprinting attack on a TBB enabled with HTTP/2 in a closed-world scenario.
2. For a website fingerprinting attack on a TBB enabled with HTTP/2 the decrease in accuracy was minimal compared to a TBB enabled with just HTTP/1.1.

Discussion & Future work

1. Closed-world scenario not realistic and experiments do not conform with human browsing habits (Juarez et al., 2014).
2. Some websites are hard to fingerprint due to: A/B testing, localisation and/or random content.
3. An attacker would need to continually keep his model up-to-date due to changing websites.
4. HTTP/2 prioritisation could be used to randomise traffic and increase fingerprinting difficulty.

Thank you for listening!

Thank you for listening!
Are there any questions?

Optimal string aligment distance

Algorithm 2 Optimal string alignment distance

Input: Strings s_1, s_2 with $|s_1| = m$ and $|s_2| = n$; insertion/deletion cost $cost_{id}$, substitution cost $cost_{sub}$, transposition cost $cost_{trans}$

Output: OSAD of s_1 and s_2

Initialize matrix M of dimensions m by n, with:
$M(i, 0) = i \cdot cost_{id} \quad \forall\, 0 \leq i \leq m$
$M(0, j) = j \cdot cost_{id} \quad \forall\, 0 < j \leq n$
for $0 < i \leq m, 0 < j \leq n$ **do**
    if $s_1(i) = s_2(j)$ **then** $cost_{idt} = 0$
    else $cost_{idt} = cost_{id}$
    end if
    $M_{ins} = M(i-1, j) + cost_{idt}$
    $M_{del} = M(i, j-1) + cost_{idt}$
    $M_{sub} = M(i-1, j-1) + cost_{sub}$
    if $s_1(i) = s_2(j-1)$ & $s_1(i-1) = s_2(j)$ **then**
        $M_{transpose} = M(i-2, j-2) + cost_{trans}$
    else
        $M_{transpose} = +\infty$
    end if
    $M(i, j) = \min\{M_{ins}, M_{del}, M_{sub}, M_{transpose}\}$
end for
Return $M(m, n)$

Figure: As in Appendix B of Wang and Goldberg (2013).

References I

"How Tor works" images on slides 7 based on "How Tor Works" images from `https://www.torproject.org/about/overview`.
Devil, Py, Coding, Monitor and Onion icons in figure on slide 8, 13 and 7 made by Freepik from www.flaticon.com and is licensed by CC 3.0 BY.
Server and Folder icons in figure on slide 13 and 7 made by Madebyoliver from www.flaticon.com and is licensed by CC 3.0 BY.

References II

Henri Bal, Dick Epema, Cees de Laat, Rob van Nieuwpoort, John Romein, Frank Seinstra, Cees Snoek, and Harry Wijshoff. A medium-scale distributed system for computer science research: Infrastructure for the long term. *Computer*, 49(5):54–63, 2016.

Xiang Cai, Xin Cheng Zhang, Brijesh Joshi, and Rob Johnson. Touching from a distance: Website fingerprinting attacks and defenses. In *Proceedings of the 2012 ACM conference on Computer and communications security*, pages 605–616. ACM, 2012.

Dominik Herrmann, Rolf Wendolsky, and Hannes Federrath. Website fingerprinting: attacking popular privacy enhancing technologies with the multinomial naïve-bayes classifier. In *Proceedings of the 2009 ACM workshop on Cloud computing security*, pages 31–42. ACM, 2009.

References III

Marc Juarez, Sadia Afroz, Gunes Acar, Claudia Diaz, and Rachel Greenstadt. A critical evaluation of website fingerprinting attacks. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, pages 263–274. ACM, 2014.

Marc Liberatore and Brian Neil Levine. Inferring the source of encrypted http connections. In *Proceedings of the 13th ACM conference on Computer and communications security*, pages 255–263. ACM, 2006.

Andriy Panchenko, Lukas Niessen, Andreas Zinnen, and Thomas Engel. Website fingerprinting in onion routing based anonymization networks. In *Proceedings of the 10th annual ACM workshop on Privacy in the electronic society*, pages 103–114. ACM, 2011.

References IV

Tao Wang and Ian Goldberg. Improved website fingerprinting on tor. In *Proceedings of the 12th ACM workshop on Workshop on privacy in the electronic society*, pages 201–212. ACM, 2013.